

SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA

LIB  
V04

VV		VV	AAAAAA	LL	IIIIII	DDDDDDDD		AAAAAA	TTTTTTTTTT	EEEEEEEEEE	
VV		VV	AAAAAA	LL	IIIIII	DDDDDDDD		AAAAAA	TTTTTTTTTT	EEEEEEEEEE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EEEEEEEE	
VV		VV	AA	AA	II	DD	DD	AA	TT	EEEEEEEE	
VV		VV	AAAAAAAAA	LL	II	DD	DD	AAAAAAAAA	TT	EE	
VV		VV	AAAAAAAAA	LL	II	DD	DD	AAAAAAAAA	TT	EE	
	VV	VV	AA	AA	II	DD	DD	AA	TT	EE	
	VV	VV	AA	AA	II	DD	DD	AA	TT	EE	
		VV	AA	AA	II	DD	DD	AA	TT	EE	
		VV	AA	AA	IIIIII	DDDDDDDD		AA	TT	EEEEEEEEEE	....
		VV	AA	AA	IIIIII	DDDDDDDD		AA	TT	EEEEEEEEEE	....

```

LL          IIIII
LL          IIIII
           I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LL          I
LLL        IIIII
LLL        IIIII

SSSSSSSSS
SSSSSSSSS
      SS
      SS
      SS
      SS
    SSSSSS
    SSSSSS
              SS
              SS
              SS
              SS
SSSSSSSSS
SSSSSSSSS

```

[illegible]

(1) 53  
(1) 81  
(2) 122

DECLARATIONS  
VAL\_SET\_MAX - Set maximum number of links to traverse  
VALIDATE\_QUEUE - Validate queue structure



```
0000 1      .TITLE  VALIDATE - Structure Validation module
0000 2      .IDENT  'V04-000'
0000 3
0000 4      *****
0000 5      *
0000 6      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      *  ALL RIGHTS RESERVED.
0000 9      *
0000 10     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     *  TRANSFERRED.
0000 16     *
0000 17     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     *  CORPORATION.
0000 20     *
0000 21     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *  *****
0000 25     *
0000 26     *
0000 27     *
0000 28     *++
0000 29     *
0000 30     *  FACILITY:
0000 31     *
0000 32     *      SDA
0000 33     *
0000 34     *  ABSTRACT:
0000 35     *
0000 36     *      This module contains code which verifies the consistency of
0000 37     *      certain VMS internal structures.
0000 38     *
0000 39     *  ENVIRONMENT:
0000 40     *
0000 41     *      VMS user mode.
0000 42     *
0000 43     *--
0000 44     *
0000 45     *  AUTHOR: Jake VanNoy, CREATION DATE: 21-Jan-1983
0000 46     *
0000 47     *  MODIFIED BY:
0000 48     *
0000 49     *      V03-000 JLV0226      Jake VanNoy      21-JAN-1983
0000 50     *      Initial coding.
0000 51     *  **
0000 52     *
0000 53     *  .SBTTL  DECLARATIONS
0000 54     *
0000 55     *  INCLUDE FILES:
0000 56     *
0000 57     *
```

```
0000 58      $OPTDEF
0000 59      $TPADEF
0000 60
0000 61 ::
0000 62 :: MACROS:
0000 63 ::
0000 64 ::
0000 65 ::
0000 66 :: EQUATED SYMBOLS:
0000 67 ::
00000000 0000 68 NEWFL = 0
00000004 0000 69 NEWBL = 4
00000008 0000 70 HEADFL = 8
0000000C 0000 71 HEADBL = 12
00000010 0000 72 LOCAL_STORAGE = 16
0000 73
0000 74 ::
0000 75 :: OWN STORAGE:
0000 76 ::
0000 77
00000190 0000 78 VALIDATE_MAX: .LONG 400
0004 79
0004 80
0004 81 .SBTTL VAL_SET_MAX - Set maximum number of links to traverse
0004 82
0004 83 ::++
0004 84 ::
0004 85 :: FUNCTIONAL DESCRIPTION:
0004 86 ::
0004 87 :: Tparse action routine which sets value of VALIDATE_MAX.
0004 88 :: *** Note that the VALIDATE_QUEUE code does not use this
0004 89 :: *** as input. It has not been proven that there are "infinite"
0004 90 :: *** queues which the algorithm does not detect.
0004 91 ::
0004 92 :: CALLING SEQUENCE:
0004 93 :: NONE
0004 94 ::
0004 95 :: INPUT PARAMETERS:
0004 96 :: NONE
0004 97 ::
0004 98 :: IMPLICIT INPUTS:
0004 99 :: NONE
0004 100 ::
0004 101 :: OUTPUT PARAMETERS:
0004 102 :: NONE
0004 103 ::
0004 104 :: IMPLICIT OUTPUTS:
0004 105 :: NONE
0004 106 ::
0004 107 :: COMPLETION CODES:
0004 108 :: NONE
0004 109 ::
0004 110 :: SIDE EFFECTS:
0004 111 :: NONE
0004 112 ::
0004 113 ::--
0004 114
```

VALIDATE  
V04-000

M 8  
- Structure Validation module  
VAL\_SET\_MAX - Set maximum number of link

16-SEP-1984 01:48:32 VAX/VMS Macro V04-00  
5-SEP-1984 03:34:48 [SDA.SRC]VALIDATE.MAR;1

Page 3  
(1)

		0000	0004	115				
			0004	116	.Entry VAL_SET_MAX, 0			
			0006	117				
FS	AF	1C	AC	D0	0006	118	MOVL	TPASL_NUMBER(AP),VALIDATE_MAX ; set max
		50	01	D0	000B	119	MOVL	#1,R0
				04	000E	120	RET	

LIB  
V04



```
000F 122      .SBTTL  VALIDATE_QUEUE - Validate queue structure
000F 123
000F 124      :++
000F 125
000F 126      FUNCTIONAL DESCRIPTION:
000F 127
000F 128      The algorithm used to validate a doubly linked list queue
000F 129      starts by copying the head (or starting point specified)
000F 130      into a known location. It then proceeds by following the
000F 131      forward link checking that the backward link points to where
000F 132      the last forward link was.
000F 133
000F 134      CALLING SEQUENCE:
000F 135      CALLS from tparse.
000F 136
000F 137      INPUT PARAMETERS:
000F 138      TPA$L_NUMBER(AP) - address to start search from
000F 139
000F 140      IMPLICIT INPUTS:
000F 141
000F 142      OPTIONS - can specify SELF RELATIVE QUEUE
000F 143      *** code not written for this as yet.
000F 144
000F 145      OUTPUT PARAMETERS:
000F 146      NONE
000F 147
000F 148      IMPLICIT OUTPUTS:
000F 149      NONE
000F 150
000F 151      COMPLETION CODES:
000F 152
000F 153      no such memory, or success
000F 154
000F 155      SIDE EFFECTS:
000F 156      NONE
000F 157
000F 158      :--
000F 159
000F 160
000F 161
00FC 000F 162 .Entry  VALIDATE_QUEUE, ^M<R2,R3,R4,R5,R6,R7>
0011 163
0011 164      MOVL  TPA$L_NUMBER(AP),R0      ; Address to start at
0015 165      MOVL  R0,ADDRESS                ; Set "current"
001C 166
001C 167      MOVL  VALIDATE_MAX,R1           ; Max number of links *** not used
001C 168
001C 169      SUBL2  #local_storage,SP         ; allocate storage from stack
001F 170      MOVL  SP,R2                    ; Allocate on stack
0022 171      CLRL  R6                        ; Counter
0024 172
0024 173      MOVL  R0,R3                      ; init last pointer
0027 174      TRYMEM (R0),HEADFL(R2),#8      ; try memory at head of queue
0035 175      MOVL  HEADFL(R2),R4             ; next address
0039 176
0039 177      ; Loop through flinks
0039 178
```

50 1C AC D0  
00000000'EF 50 D0

5E 10 C2  
52 5E D0  
56 D4

53 50 D0  
54 08 A2 D0

```
0039 179 20$:
0039 180 TRYMEM (R4),NEWFL(R2),#8 ; try memory
0046 181 BLBC R0,mem_err ; Error
08 A2 37 50 E9 0049 182 CMPL newfl(R2),headfl(R2) ; Same as listhead?
62 10 13 004D 183 BEQL 100$ ; Done with flinks
56 04 A2 53 D6 004F 184 INCL R6 ; Increment counter
53 53 3E 12 0051 185 CMPL R3,NEWBL(R2) ; back link ok?
54 54 D0 0055 186 BNEQ bad_blink ; branch if not
54 62 D0 0057 187 MOVL R4,R3 ; save last pointer
DA 11 005A 188 MOVL NEWFL(R2),R4 ; move to next element
005D 189 BRB 20$ ; Loop
005F 190
005F 191 ; Search completed successfully, do final validation
005F 192
0C A2 04 A2 D1 005F 193 100$: CMPL NEWBL(R2),HEADBL(R2) ; Same as listhead?
2F 12 0064 194 BNEQ bad_blink ; Done with list
0C A2 53 D1 0066 195 CMPL R3,HEADBL(R2) ; does this check out?
29 12 006A 196 BNEQ bad_blink
006C 197
006C 198 ; Queue is ok, check for empty queue
006C 199
56 D5 006C 200 110$: TSTL R6
47 12 006E 201 BNEQ queue_ok
0046 31 0070 202 PRINT 0,<The queue is empty> ;
0080 203 BRW VAL_Q_EXIT
0080 204
54 DD 0080 205 mem_err: PUSHL R4
0082 206 SIGNAL 1,NOTINPHYS ; Not in physical memory error
04 0094 207 RET
0095 208
0095 209 bad_blink:
53 DD 0095 210 PUSHL R3
0097 211 PRINT 1,-
0097 212 <Error comparing backward link to previous structure address (!XL)>
56 DD 00A4 213 PUSHL R6
54 DD 00A6 214 PUSHL R4
00A8 215 PRINT 2,-
00A8 216 <Error occured in queue element at address !XL, after tracing !UL element!%S>
0F 11 00B5 217 brb val_q_exit
00B7 218
00B7 219 queue_ok:
56 DD 00B7 220 PUSHL R6 ; Count
00B9 221 PRINT 1, <Queue is complete, total of !UL element!%S in the queue>
00C6 222
00C6 223 val_q_exit:
50 01 D0 00C6 224 MOVL #1,R0
04 00C9 225 RET
00CA 226
00CA 227
00CA 228 .END ; VALIDATE
```



VALIDATE  
Symbol table

- Structure Validation module

C 9

16-SEP-1984 01:48:32 VAX/VMS Macro V04-00  
5-SEP-1984 03:34:48 [SDA.SRC]VALIDATE.MAR;1Page 6  
(2)

ADDRESS	*****	X	01
ARGS	= 00000003		
BAD_BLINK	00000095	R	01
HEADBL	= 0000000C		
HEADFL	= 00000008		
LIBSSIGNAL	*****	X	01
LOCAL_STORAGE	= 00000010		
MEM_ERR	00000080	R	01
MSG\$ NOTINPHYS	*****	X	01
NEWBC	= 00000004		
NEWFL	= 00000000		
PRINT	*****	X	01
QUEUE_OK	000000B7	R	01
TPASL_NUMBER	= 0000001C		
TRYMEM	*****	X	01
VALIDATE_MAX	00000000	R	01
VALIDATE_QUEUE	0000000F	RG	01
VAL_Q_EXIT	000000C6	R	01
VAL_SET_MAX	00000004	RG	01

+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	000000CA ( 202.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS\$	00000000 ( 0.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
LITERALS	000000F5 ( 245.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:00.80
Command processing	111	00:00:00.48	00:00:02.85
Pass 1	163	00:00:01.68	00:00:06.93
Symbol table sort	0	00:00:00.12	00:00:00.12
Pass 2	53	00:00:00.47	00:00:02.15
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	363	00:00:02.84	00:00:12.89

The working set limit was 1050 pages.

13151 bytes (26 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 182 non-local and 8 local symbols.

228 source lines were read in Pass 1, producing 21 object records in Pass 2.

13 pages of virtual memory were used to define 12 macros.

-----  
! Macro library statistics !  
-----

Macro library name

Macros defined

-----  
\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1  
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1  
\$255\$DUA28:[SYSLIB]STARLET.MLB;2  
TOTALS (all libraries)

5  
0  
4  
9

275 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:VALIDATE/OBJ=OBJ\$:VALIDATE MSRC\$:VALIDATE/UPDATE=(ENH\$:VALIDATE)+EXECMLS/LIB+LIB\$:SDALIB/LIB



0354 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

SYMBOLS  
LIS

SMGRTL

SMGBLDTRM  
MAP

SDAMSG  
LIS

VAXINST  
LIS

SMGMAPTRM  
MAP

SMGKCB  
SDL

VALIDATE  
LIS

STACKS  
LIS

SMGDEF  
SDL

SMGKDE  
SDL

SMGSHR  
MAP